REMARKS

Claims 1, 2, 5, 7, 9, 12, 13 – 16, 19 – 22, 24, 27, 30 – 33, and 35 have been amended. Claims 36 and 37 have been cancelled. Hence, claims 1-24 and 26-35 are pending in the application.

Claims 1 – 24 and 26 - 35 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,560,005, herein *Hoover*, in view of U.S. Patent No. 5,809,297, issued to David M. Kroenke, et al., herein *Kroenke*. These rejections are traversed.

DESCRIPTION OF CITED ART

Because the rejections were based on *Hoover*, a description of *Hoover* is useful. *Hoover* describes a technique for managing and locating objects used to model relational data from customer databases. From each customer database, relational data is imported into a remote database associated with the customer database, and in particular, into objects stored in the remote database. The attributes of the objects are stored in object attribute tables. (Column 6, lines 41-56, Column 27, lines 11-18, line 27-36, Column 38, lines *30-50).* Each object is associated with an object identifier. (Column 13, lines 36-39, Column 22, lines 17-33). Attributes of an object may be replicated, that is, a particular attribute of an object may be stored in an object attribute table on multiple remote databases. (Column 24, lines 44-60, Column 32, lines 118). An object attribute table that stores data for an object does not necessarily contain data for all the attributes of the object, or the most current data for an attribute (Id., col. 27, lines 49 - 55).

An object broker maintains a map table. The object broker uses the map table to locate the object attribute tables that contain attributes for an object. The map table maps an object identifier to an object attribute table on one or more remote databases that contain the

object. (Column 6, lines 33-42, lines *52-55*, Column 24, lines 2-60, Column *35*, lines 7-19).

In addition, the object broker maintains object index tables that map key values (i.e. attributes) to object identifiers. (Column *25*, lines 10-15). The object broker uses the object index table to find an object identifier mapped to a key value, and uses the object identifier to find the object attribute tables and the locations mapped to the object identifier. (Column *53*, lines 8-48).

Object identifiers are permanently assigned to objects when the objects are created in the object attributes tables. (Column 22, lines 4146). Specifically, each remote database is assigned a range of unique numbers. As objects are created on a remote database system, each object is assigned a permanent object identifier from the unique range assigned to the remoter database system. (id.) The object identifiers are sequentially assigned, and are not in any way based on the attribute of the object being assigned an object identifier.

Because the cited art is based on *Kroenke*, a description of *Kroenke* is useful. The following are excerpts from *Kroenke*.

> Transformation of Semantic Objects into Relational Tables
>
> As described above, the present invention is a system that allows a user to create a semantic object data model that represents data to be stored in a database. The system then transforms the model into one or more of a set of commands that are interpreted by a conventional database program to create a plurality of relational database tables that correspond to the semantic object data model. (col. 27, line 63 to col. 28, line 4) The particular table definitions produced by the present invention can be tailored to a particular database protocol determined based upon the user's election of a specific commercial database program with which the present invention will be used. Given the following description of how the semantic object model is transformed into a number of relational tables, it is considered to be within the skill of a computer programmer to create a driver for any of the commercially available relational database programs that will produce the appropriate set of commands that cause that

database to create corresponding relational database tables for the desired schema. (col. 28, line 32 – 43)

CLAIMS 1, 16, AND 20

Claims 1 and 20 recite:

reading data from one or more rows of the set of one or more tables, wherein said one or more rows do not store an object id used for modeling the data in said one or more rows as an object that belongs to said object class;

reading database metadata that defines how to derive object ids from values in one or more columns;

generating, in the manner defined by said database metadata, an object id derived from one or more values in the one or more columns in said one or more rows;....

Claim 16 recites:

database metadata that indicates how to derive object ids from values in one or more columns;

a set of one or more tables, said set of one or more tables containing one or more rows, wherein said one or more rows do not store an object id used for modeling the data in said one or more rows as an object that belongs to said object class;

said processor configured to read data from one or more rows of the set of one or more tables;

said processor configured to generate, in the manner defined by said database metadata, an object id derived from one or more values from said one or more rows, wherein said one or more values reside in said one or more columns;...

Claims 1, 16, and 20 contain features not disclosed or suggested by the cited art. Among these features is to generate, in the manner defined by database metadata, an object id derived from one or more values in one or more columns. In *Hoover*, object ids are generated by assigning an identifier from unique range of identifiers allocated to a remote database system. Nothing about this way of generating object ids suggests generating an object id from values in columns, let alone generating them in the manner defined by database metadata.

In rejecting claims 1, 16, and 20, the Office Action cites col. 29, lines 39 through col. 30, line 5 as disclosing or suggesting generating an object id as claimed. This section discusses storing object ids in tables and indexes. Perhaps the Office Action is alleging that storing object ids in the rows of a table or index, which can be defined by database metadata, suggests generating an object id, based on database metadata, because the objects ids may be later accessed in the rows of the table or index for some purpose, such as a look-up. However, claims 1, 16, and 20 expressly require that any object id generated as claimed be generated from rows that do not store an object id used for modeling data as objects. Thus, at best *Hoover* teaches that object ids used to model data are stored in the rows of tables and indexes. *Hoover*, however, does not teach that object ids are generated from rows that do not contain such object ids, as claimed.

Finally, *Kroenke* does not disclose or suggest the features of claims 1, 16, and 20 discussed above. Therefore, claims 1, 16, and 20 are patentable. Reconsideration and allowance of claims 1, 16 and 20 is respectfully requested.

CLAIM 7, 19, AND 24

Claims 7 and 24 recite:

reading first database metadata that indicates how to generate a column object from one or more columns;

reading a first set of data from the one or more columns of a plurality of rows from the set of one or more tables, wherein second database metadata defines said one or more tables, wherein said second database metadata does not specify how to generate said column object from said one or more columns;

generating, in the manner indicated by said first database metadata, a particular column object based on said first set of data;....

Claim 19, recites:

first database metadata that defines how to generate a column object from one or more columns;

second database metadata that defines said one or more tables, wherein said second database metadata does not specify how to generate said column object from said one or more columns;

said processor configured to read a first set of data from a plurality of rows from the set of one or more tables;

said processor configured to generate, in the manner defined by said first database metadata, a particular column object based on said first set of data...

Claims 7, 19, and 24 contain features not disclosed or suggested in any way by the cited art. Among them is (1) to generate a column object based on data from MULTIPLE rows from a table, and (2) to generate the column object, in a manner defined by database metadata other than database metadata that defines the table, where the other database metadata also does not specify how to generate the column object from one or more columns. Applicant has searched *Hoover* and *Kroenke* and has not even found a teaching about column objects or how they are generated. Therefore, the cited art cannot disclose or suggest generating a column object, let alone the particular way claimed for generating a column object. Reconsideration and allowance of claims 7, 19, and 24 is respectfully requested.

DEPENDENT CLAIMS

The remainder of the claims are dependant claims, each of which depend (directly or indirectly) on one of the claims discussed above. Each of the dependant claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of the dependant claims introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those limitations is not included at this time.

It is respectfully requested that the Examiner reconsider all of the pending claims, which are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

Throughout the pendency of this application, please charge any fees to deposit account 50-1302.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: November 21, 2002

Marcel K. Bingham
Reg. No. 42,327

1600 Willow Street
San Jose, CA 95125
Telephone No.: (408) 414-1080 ext.206
Facsimile No.: (408) 414-1076

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office Fax No. (703) 872-9326

on 11/21/02 by Trudy Bagdon

50277-0370
OID-1997-10-09-DIV

MARKED-UP VERISON OF CLAIMS

1. (Amended) A method for presenting data from a set of one or more tables as a set of objects that belong to an object class, the method comprising the steps of:

reading data from one or more rows of the set of one or more tables, wherein said one or more rows do not store an object id used for modeling the data in said one or more rows as an object that belongs to said object class;

reading database metadata that defines how to derive object ids from values in one or more columns;

generating ~~based on~~, in the manner defined by said database metadata, an object id derived from one or more values ~~from~~ in the one or more columns in said one or more rows~~, wherein said one or more values reside in one or more columns, wherein said database metadata specifies that said object id is generated based on values in said one or more columns~~; and

presenting data from said one or more rows as an object that belongs to said object class and that has said object id.

2. (Amended) The method of Claim 1, wherein the step of generating ~~based on database metadata~~ an object id includes generating an object id based on values from one or more rows of a relational table that belongs to the set of one or more tables.

3. (Unchanged) The method of Claim 1, further comprising the step of generating a reference to the object based on the object id.

4. (Unchanged) The method of Claim 3, further comprising the step of accessing the object based on the reference generated for the object.

5. (Amended) The method of Claim 1, wherein:
the method further includes the steps of:
receiving a request to define a view, said request specifying that said object id is generated based on values in said one or more columns;
in response to receiving the request to define the view, storing specification data that specifies the one or more columns; and
the step of generating ~~based on database metadata~~ an object id includes determining how to generate the object id by inspecting said specification data.

6. (Unchanged) The method of Claim 5, wherein the step of receiving a request to define a view includes receiving a request that specifies the one or more columns as including at least one column from a relational table.

7. (Amended) A method for presenting, as an object, data from a set of one or more tables residing in one or more databases, the method comprising the steps of:
reading first database metadata that indicates how to generate a column object from one or more columns;
reading a first set of data from the one or more ~~fields~~ columns of a plurality of rows from the set of one or more tables, wherein second database metadata defines

said one or more tables, wherein said second database metadata does not specify how to generate said column object from said one or more columns;

generating ~~a~~, in the manner indicated by said first database metadata, a particular column object based on said first set of data; and

presenting a second set of data from said set of one or more tables as object oriented data by presenting said second set of data as said object that has said particular column object as an attribute.

8. (Unchanged) The method of Claim 7, wherein the step of reading a first set of data includes reading data from one or more rows of at least one relational table.

9. (Amended) The method of Claim 7, wherein the step of generating a particular column object includes generating a collection object.

10. (Unchanged) The method of Claim 9, wherein the step of generating a collection object includes generating said collection object as a list of elements belonging to a single data type.

11. (Unchanged) The method of Claim 9, wherein the step of generating a collection object includes generating said collection object as a nested table.

12. (Amended) The method of Claim 9, wherein the step of generating a particular column object includes generating a column object belonging to a user specified object type.

13. (Amended) The method of Claim 9, where the step of generating a particular column object includes generating a column object that is a reference to another object.

14. (Amended) The method of Claim 13, wherein the step of generating a particular column object includes generating a column object that is a reference to an object presented by an object view.

15. (Amended) The method of Claim 13, wherein the step of generating a particular column object includes generating a column object that is a reference to an object residing in a database.

16. (Amended) A computer system, comprising:

a processor;

a memory coupled to said processor;

database metadata that indicates how to derive object ids from values in one or more columns;

a set of one or more tables, said set of one or more tables containing one or more rows, wherein said one or more rows do not store an object id used for

modeling the data in said one or more rows as an object that belongs to said object class;

said processor configured to read data from one or more rows of the set of one or more tables;

said processor configured to generate ~~based on~~, in the manner defined by said database metadata, an object id derived from one or more values from said one or more rows, wherein said one or more values reside ~~in one or more columns, wherein said database metadata specifies that said object id is generated based on values~~ in said one or more columns; and

said processor configured to present data from said one or more rows as an object that belongs to an object class and that has said object id.

17. (Unchanged) The computer system of Claim 16, wherein said values from said one or more rows include values from one or more rows of a relational table that belongs to said set of one or more tables.

18. (Unchanged) The computer system of Claim 16, further comprising:

said processor configured to receive a request to define a view, said request specifying one or more columns of the set of one or more tables containing values used to generate said object id;

said processor configured to respond to receiving the request to define the view by storing specification data that specifies the one or more columns; and

said processor configured to generate the object id based on values from said one or more rows by determining how to generate the object id by inspecting said specification data.

19. (Amended) A computer system, comprising:

a processor;

a memory coupled to said processor;

one or more databases;

a set of one or more tables contained in said one or more databases;

first database metadata that defines how to generate a column object from one or more columns;

second database metadata that defines said one or more tables, wherein said second database metadata does not specify how to generate said column object from said one or more columns;

said processor configured to read a first set of data from a plurality of rows from the set of one or more tables;

said processor configured to generate ~~a~~, in the manner defined by said first database metadata, a particular column object based on said first set of data; and

said processor configured to present a second set of data from said set of one or more tables as object oriented data by presenting said second set of data as said object that has said particular column object as an attribute.

20. (Amended) A computer-readable medium carrying one or more sequences of one or

more instructions for presenting data from a set of one or more tables as a set of objects that belong to an object class, wherein the execution of the one or more sequences of the one or more instructions causes the one or more processors to perform the steps of:

reading data from one or more rows of the set of one or more tables, wherein said one or more rows do not store an object id used for modeling the data in said one or more rows as an object that belongs to said object class;

reading database metadata that defines how to derive object ids from values in one or more columns;

generating ~~based on~~, in the manner defined by said database metadata, an object id derived from one or more values ~~from~~ in the one or more columns in said one or more rows~~, wherein said one or more values reside in one or more columns, wherein said database metadata specifies that said object id is generated based on values in said one or more columns~~; and

presenting data from said one or more rows as an object that belongs to said object class and that has said object id.

21. (Amended) The computer readable medium of Claim 20, wherein the step of generating ~~based on database metadata~~ an object id includes generating an object id based on values from one or more rows of a relational table that belongs to the set of one or more tables.

22. (Amended) The computer readable medium of Claim 21, wherein:

the one or more sequences of instructions includes one or more instructions for performing the steps of:

receiving a request to define a view, said request specifying one or more columns of the set of one or more tables containing values used to generate said object id;

in response to receiving the request to define the view, storing specification data that specifies the one or more columns; and

the step of generating an object id ~~based on values from said one or more rows~~ includes determining how to generate the object id by inspecting said specification data.

23. (Unchanged) The computer readable medium of Claim 22, wherein the step of receiving a request to define a view includes receiving a request that specifies the one or more columns as including at least one column from a relational table.

24. (Amended) A computer-readable medium carrying one or more sequences of one or more instructions for presenting, as an object, data from a set of one or more tables residing in one or more databases, wherein the execution of the one or more sequences of the one or more instructions causes the one or more processors to perform the steps of:

reading <u>first database metadata that defines how to generate a column object from one or more columns;</u>

reading a first set of data from the one or more ~~fields~~ columns of a plurality of rows from the set of one or more tables, wherein second database metadata defines said one or more tables, wherein said second database metadata does not specify how to generate said column object from said one or more columns;

generating ~~a~~, in the manner defined by said first database metadata, a particular column object based on said first set of data; and

presenting a second set of data from said set of one or more tables as object oriented data by presenting said second set of data as said object that has said particular column object as an attribute.

25. Cancelled

26. (Unchanged) The computer-readable medium of Claim 24, wherein the step of reading a first set of data includes reading data from one or more rows of at least one relational table.

27. (Amended) The computer-readable medium of Claim 24, wherein the step of generating a particular column object includes generating a collection object.

28. (Unchanged) The computer-readable medium of Claim 27, wherein the step of generating a collection object includes generating said collection object as a list of elements belonging to a single data type.

29. (Unchanged) The computer-readable medium of Claim 27, wherein the step of generating a collection object includes generating said collection object as a nested table.

30. (Amended) The computer-readable medium of Claim 27, wherein the step of generating a particular column object includes generating a column object belonging to a user specified object type.

31. (Amended) The computer-readable medium of Claim 27, wherein the step of generating a particular column object includes generating a column object that is a reference to another object.

32. (Amended) The computer-readable medium of Claim 31, wherein the step of generating a particular column object includes generating a column object that is a reference to an object presented by an object view.

33. (Amended) The computer-readable medium of Claim 31, wherein the step of generating a particular column object includes generating a column object that is a reference to an object residing in a database.

34. (Unchanged) The computer-readable medium of Claim 20, the steps further comprising the step of generating a reference to the object based on the object id.

35. (Amended) The computer-readable medium of Claim ~~20~~ 34, wherein the steps further comprise the step of accessing the object based on the reference generated for the object.

36. Cancelled

37. Cancelled